

PATENT APPLICATION

**METHOD AND APPARATUS FOR A THIN
CELP VOICE CODEC**

Inventor: Marwan A. Jabri, a citizen of Australia, residing at
Level 7, 3 Small Street
Broadway, NSW, 2007 Australia

Nicola Chong-White, a citizen of New Zealand, residing at
364 Penshurst Street
Chatswood, NSW, 2067 Australia

Jianwei Wang, a citizen of Australia, residing at
104 Killarney Drive
Killarney Heights, NSW, 2087 Australia

Assignee: Dilithium Holdings, Inc.
700 Larkspur Landing Circle, Suite 263,
Larkspur, CA, 94939

Entity: Small

METHOD AND APPARATUS FOR A THIN CELP VOICE CODEC

CROSS-REFERENCES TO RELATED APPLICATIONS

5 **[0001]** This application claims priority to U.S. Provisional Nos. 60/419776 filed 10/17/2002 and 60/439366 filed 01/09/2003, which are incorporated by reference herein.

STATEMENT AS TO RIGHTS TO INVENTIONS MADE UNDER FEDERALLY SPONSORED RESEARCH OR DEVELOPMENT

[0002] NOT APPLICABLE

10

BACKGROUND OF THE INVENTION

[0003] The present invention relates generally to telecommunication techniques. More particularly, the invention provides an encoding and decoding system and method that support a plurality of compression standards and share computational resources. Merely by way of example, the invention has been applied to Code Excited Linear Prediction (CELP) techniques, but it would be recognized that the invention has a much broader range of applicability.

15 **[0004]** Code Excited Linear Prediction (CELP) speech coding techniques are widely used in mobile telephony, voice trunking and routing, and Voice-over-IP (VoIP). Such coders/decoders (codecs) model voice signals as a source filter model. The source/excitation signal is generated via adaptive and fixed codebooks, and the filter is modeled by a short-term linear predictive coder (LPC). The encoded speech is then represented by a set of parameters which specify the filter coefficients and the type of excitation.

20 **[0005]** Industry standards codecs using CELP techniques include Global System for Mobile (GSM) Communications Enhanced Full Rate (EFR) codec, Adaptive Multi-Rate Narrowband (AMR-NB) codec, Adaptive Multi-Rate Wideband (AMR-WB), G.723.1, G.729, Enhanced Variable Rate Codec (EVRC), Selectable Mode Vocoder (SMV), QCELP, and MPEG-4. These standard codecs apply substantially the same generic algorithms in

extracting CELP parameters with modifications to frame and subframe sizes, filtering procedures, interpolation resolutions, code-book structures and code-book search intervals.

[0006] For example, the GSM standards AMR-NB and AMR-WB usually operate with a 20ms frame size divided into 4 subframes of 5ms. One difference between the wideband and narrowband coder is the sampling rate, which is 8 kHz for AMR-NB and 16 kHz downsampled to 12.8 kHz for analysis for AMR-WB. The linear prediction (LP) techniques used in both AMR-NB and AMR-WB are substantially identical, but AMR-WB performs adaptive tilt filtering, linear prediction (LP) analysis to 16th order over an extended bandwidth of 6.4 kHz, conversion of LP coefficients to/from Immittance Spectral Pairs (ISP), and quantization of the ISPs using split-multi-stage vector quantization (SMSVQ). The pitch search routines and computation of the target signal are similar. Both codecs follow an ACELP fixed codebook structure using a depth-first tree search to reduce computations. The adaptive and fixed codebook gains are quantized in both codecs using joint vector quantization (VQ) with 4th order moving average (MA) prediction. AMR-WB also contains additional functions to deal with the higher frequency band up to 7 kHz.

[0007] In another example, the Code Division Multiple Access (CDMA) standards SMV and EVRC share certain math functions at the basic operations level. At the algorithm level, the noise suppression and rate selection routines of EVRC are substantially identical to SMV modules. The LP analysis follows substantially the same algorithm in both codecs and both modify the target signal to match an interpolated delay contour. At Rate 1/8, both codecs produce a pseudo-random noise excitation to represent the signal. SMV incorporates the full range of post-processing operations including tilt compensation, formant postfilter, long term postfilter, gain normalization, and highpass filtering, whereas EVRC uses a subset of these operations.

[0008] As discussed above, a large number of industry standards codecs use CELP techniques. These codecs are usually supported by mobile and telephony handsets in order to interoperate with emerging and legacy network infrastructure. With the deployment of media rich handsets and the increasing complexity of user applications on these handsets, the large number of codecs is putting increasing pressure on handset resources in terms of program memory and DSP resources.

[0009] Hence it is desirable to improve codec techniques.

BRIEF SUMMARY OF THE INVENTION

[0010] The present invention relates generally to telecommunication techniques. More particularly, the invention provides an encoding and decoding system and method that support a plurality of compression standards and share computational resources. Merely by way of example, the invention has been applied to Code Excited Linear Prediction (CELP) techniques, but it would be recognized that the invention has a much broader range of applicability.

[0011] According to an embodiment, the present invention provides a method and apparatus for encoding and decoding a speech signal using a multiple codec architecture concept that supports several CELP voice coding standards. The individual codecs are combined into an integrated framework to reduce the program size. This integrated framework is referred to as a thin CELP codec. The apparatus includes a CELP encoder that generates a bitstream from the input voice signal in a format specific to the desired CELP codec, and a CELP decoding module that decodes a received CELP bitstream and generates a voice signal. The CELP encoder includes one or more codec-specific CELP encoding modules, a common functions library, a common math operations library, a common tables library, and a bitstream packing module. The common libraries are shared between more than one voice coding standard. The output bitstream may be bit-exact to the standard codec implementation or produce quality equivalent to the standard codec implementation. The CELP decoder includes bitstream unpacking module, one or more codec-specific CELP decoding modules, a common functions library, a common math operations library and a library of common tables. The output voice signal may be bit-exact to the standard codec implementation or produce quality equivalent to the standard codec implementation.

[0012] According to another embodiment, the method for encoding a voice signal includes generating CELP parameters from the input voice signal in a format specific to the desired CELP codec and packing the codec-specific CELP parameters to the output bitstream. The method for decoding a voice signal includes unpacking the bitstream into codec-specific CELP parameters, and decoding the parameters to generate output speech.

[0013] According to yet another embodiment of the present invention, an apparatus for encoding and decoding a voice signal includes an encoder configured to generate an output bitstream signal from an input voice signal. The output bitstream signal is associated with at least a first standard of a first plurality of CELP voice compression standards. Additionally,

the apparatus includes a decoder configured to generate an output voice signal from an input bitstream signal. The input bitstream signal is associated with at least a first standard of a second plurality of CELP voice compression standards. The CELP encoder includes a plurality of codec-specific encoder modules. At least one of the plurality of codec-specific encoder modules including at least a first table, at least a first function or at least a first operation. The first table, the first function or the first operation is associated with only a second standard of the first plurality of CELP voice compression standards. Additionally, the CELP encoder includes a plurality of generic encoder modules. At least one of the plurality of generic encoder modules includes at least a second table, a second function or a second operation. The second table, the second function or the second operation is associated with at least a third standard and a fourth standard of the first plurality of CELP voice compression standards. The third standard and the fourth standard of the first plurality of CELP voice compression standards are different. The CELP decoder includes a plurality of codec-specific decoder modules. At least one of the plurality of codec-specific decoder modules includes at least a third table, at least a third function or at least a third operation. The third table, the third function or the third operation is associated with only a second standard of the second plurality of CELP voice compression standards. Additionally, the CELP decoder includes a plurality of generic decoder modules. At least one of the plurality of generic decoder modules includes at least a fourth table, a fourth function or a fourth operation. The fourth table, the fourth function or the fourth operation is associated with at least a third standard and a fourth standard of the second plurality of CELP voice compression standards. The third standard and the fourth standard of the second plurality of CELP voice compression standards are different.

[0014] According to yet another embodiment of the present invention, a method for encoding and decoding a voice signal includes receiving an input voice signal, processing the input voice signal, and generating an output bitstream signal based on at least information associated with the input voice signal. The output bitstream signal is associated with at least a first standard of a first plurality of CELP voice compression standards. Additionally, the method includes receiving an input bitstream signal, processing the input bitstream signal, and generating an output voice signal based on at least information associated with the input bitstream signal. The output voice signal is associated with at least a first standard of a second plurality of CELP voice compression standards. The processing the input voice signal uses at least a first common functions library, at least a first common math operations library,

and at least a first common tables library. The first common functions library includes a first function; the first common math operations library includes a first operation, and the first common tables library includes a first table. The first function, the first operation and the first table are associated with at least a second standard and a third standard of the first plurality of CELP voice compression standards. The second standard and the third standard of the first plurality of CELP voice compression standards are different. The generating an output bitstream signal includes generating a first plurality of codec-specific CELP parameters based on at least information associated with the input voice signal, and packing the first plurality of codec-specific CELP parameters to the output bitstream signal. The processing the input bitstream signal uses at least a second common functions library, at least a second common math operations library, and a second common tables library. The second common functions library includes a second function, the second common math operations library includes a second operation, and the second common tables library includes a second table. The second function, the second operation and the second table are associated with at least a second standard and a third standard of the second plurality of CELP voice compression standards. The second standard and the third standard of the second plurality of CELP voice compression standards are different. The generating an output voice signal includes unpacking the input bitstream signal and decoding a second plurality of codec-specific CELP parameters to produce an output voice signal.

20 **[0015]** An example of the invention are provided, specifically a thin CELP codec which combines the voice coding standards of GSM-EFR, GSM AMR-NB and GSM AMR-WB. Another example illustrates the combination of the EVRC and SMV voice coding standards for CDMA. Many variations of voice coding standard combinations are applicable.

25 **[0016]** Numerous benefits are achieved using the present invention over conventional techniques. Certain embodiments of the present invention can be used to reduce the program size of the encoder and decoder modules to be significantly less than the combined program size of the individual voice compression modules. Some embodiments of the present invention can be used to produce improved voice quality output than the standard codec implementation. Certain embodiments of the present invention can be used to produce lower computational complexity than the standard codec implementation. Some embodiments of the present invention provide efficient embedding of a number of standard codecs and facilitates interoperability of handsets with diverse networks.

30

[0017] Depending upon the embodiment under consideration, one or more of these benefits may be achieved. These benefits and various additional objects, features and advantages of the present invention can be fully appreciated with reference to the detailed description and accompanying drawings that follow.

5

BRIEF DESCRIPTION OF THE DRAWINGS

[0018] Figures 1A and 1B are simplified illustrations of the encoder and decoder modules for voice coding to encode to and decode from multiple voice coding standards;

10 [0019] Figure 2 is a simplified diagram for a thin codec according to one embodiment of the present invention;

[0020] Figure 3 is a simplified diagram for certain parameters common to some CELP codec standards according to an embodiment of the present invention;

[0021] Figure 4 is a simplified block diagram of a CELP decoder;

[0022] Figure 5 is a simplified diagram for processing modules of a CELP encoder;

15 [0023] Figure 6 is a simplified diagram for processing modules of a CELP decoder;

[0024] Figure 7 is a simplified diagram comparing the structure of multiple individual encoders and the encoder part of a thin codec architecture according to one embodiment of the present invention;

20 [0025] Figure 8 is a simplified diagram comparing the structure of multiple individual decoders and the decoder part of a thin codec architecture according to one embodiment of the present invention;

[0026] Figure 9 is a simplified block diagram for an encoder of a thin CELP codec according to an embodiment of the present invention;

25 [0027] Figure 10 is a simplified block diagram for a decoder of a thin CELP codec according to an embodiment of the present invention;

[0028] Figure 11A is a simplified diagram showing generic modules between codec 1, codec 2 and code 3 for bit-exact implementation according to an embodiment of the present invention;

[0029] Figure 11B is a simplified diagram showing generic modules between codec 1, codec 2 and code 3 for equivalent performance implementation according to an embodiment of the present invention;

[0030] Figure 12 is a simplified block diagram of an encoder for GSM-EFR and AMR-NB;

5 [0031] Figure 13 is a simplified block diagram of an encoder for GSM AMR-WB;

[0032] Figure 14 is a simplified block diagram for an encoder of a thin codec for GSM-EFR, AMR-NB and AMR-WB according to an embodiment of the present invention;

[0033] Figure 15 is a simplified block diagram for an decoder of a thin codec for GSM-EFR, AMR-NB and AMR-WB according to an embodiment of the present invention;

10 [0034] Figure 16 is a simplified block diagram for an encoder for EVRC;

[0035] Figure 17 is a simplified block diagram of the encoder for SMV;

[0036] Figure 18 is a simplified block diagram of an embodiment of an encoder of a thin codec for SMV and EVRC according to an embodiment of the present invention.

15 [0037] Figure 19 is a simplified block diagram of an embodiment of an decoder of a thin codec for SMV and EVRC according to an embodiment of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

[0038] The present invention relates generally to telecommunication techniques. More particularly, the invention provides an encoding and decoding system and method that
20 support a plurality of compression standards and share computational resources. Merely by way of example, the invention has been applied to Code Excited Linear Prediction (CELP) techniques, but it would be recognized that the invention has a much broader range of applicability.

[0039] An illustration of the encoder and decoder modules for voice coding to encode to
25 and decode from multiple voice coding standards are shown in Figure 1A and Figure 1B. A separate encoder and decoder may be used for each coding standard, which may lead to large combined program memory requirements. Since many voice coding standards presently used are based on the Code Excited Linear Prediction (CELP) algorithm, there are many similarities in the processing functions across different coding standards.

[0040] Figure 2 is a simplified diagram for a thin codec according to one embodiment of the present invention. This diagram is merely an example, which should not unduly limit the scope of the present invention. One of ordinary skill in the art would recognize many variations, alternatives, and modifications. The thin codec 200 can encode voice samples into one of several voice compression formats, and decode bitstreams in one of several voice compression formats back to voice samples. The thin codec 200 includes an encoder system 210 and a decoder system 220. The encoder system 210 can encode the input voice samples into one of several CELP voice compression formats and the decoder system 220 can decode a bitstream in one of several CELP voice compression formats back to speech samples using an integrated codec architecture.

[0041] Figure 3 is a simplified diagram for certain parameters common to some CELP codec standards according to an embodiment of the present invention. This diagram is merely an example, which should not unduly limit the scope of the present invention. One of ordinary skill in the art would recognize many variations, alternatives, and modifications.

The intermediate parameters of open-loop pitch lag and excitation signal are usually generic to CELP codecs. The unquantized values for linear prediction parameters, pitch lags, and pitch gains are also usually generic CELP parameters. The quantized values for linear prediction parameters, adaptive codebook lags, adaptive codebook gains, fixed codebook indices, fixed codebook gains and other parameters are usually considered codec-specific parameters. For example, the quantized values for linear prediction parameters include line spectral frequencies obtained from a vector-quantization codebook.

[0042] Figure 4 is a simplified block diagram of a CELP decoder. A fixed codebook index 410 and an adaptive codebook lag 420 are used to extract vectors from a fixed codebook 412 and an adaptive codebook 422 respectively. The selected fixed codebook vector and adaptive codebook vector are gain-scaled using a decoded fixed codebook gain 414 and an adaptive codebook gain 424 respectively, and then added together to form an excitation signal 430. The excitation signal 430 is filtered by a linear prediction synthesis filter 440 to provide the spectral shape, and the resulting signal is post-processed by a post processing unit 450 to form an output speech 460.

[0043] Figure 5 is a simplified diagram for processing modules of a CELP encoder. An input speech sample 510 is first pre-processed by a pre-processing module 520. The output of the pre-processing module 520 is further processed by a linear prediction analysis and

quantization module 530. The open-loop pitch lag, adaptive codebook lag, and adaptive codebook gain are then determined and quantized by modules 540, 550, and 560 respectively. The fixed codebook indices and fixed codebook gain are then determined and quantized by modules 570 and 580 respectively. Lastly, the bitstream is packed in a desired format by a module 590.

[0044] Figure 6 is a simplified diagram for processing modules of a CELP decoder. A codec bitstream 610 is first unpacked to yield the CELP parameters by a module 620, and the excitation is reconstructed using the adaptive codebook parameters and fixed codebook parameters by a module 630. The excitation is then filtered by a linear prediction synthesis filter 640, and finally post-processing operations are applied by a module 650 to produce an output speech sample 660.

[0045] Figure 7 is a simplified diagram comparing the structure of multiple individual encoders and the encoder part of a thin codec architecture according to one embodiment of the present invention. This diagram is merely an example, which should not unduly limit the scope of the present invention. One of ordinary skill in the art would recognize many variations, alternatives, and modifications. In the thin codec architecture, individual encoders 710 are integrated into a combined codec architecture 720. Each processing module of the encoders 710 is factorized into a generic part and a specific part in the combined codec architecture 720. The program memory for the generic coding part can be shared between several voice coding standards, resulting in smaller overall program size. Depending on the bitstream constraints, the number of codecs combined, and the similarity between the codecs combined, the encoder part 720 of the thin codec may achieve significant program size reductions. The bitstream constraints may include bit-exactness and minimum performance requirements.

[0046] Figure 8 is a simplified diagram comparing the structure of multiple individual decoders and the decoder part of a thin codec architecture according to one embodiment of the present invention. This diagram is merely an example, which should not unduly limit the scope of the present invention. One of ordinary skill in the art would recognize many variations, alternatives, and modifications. In the thin codec architecture, individual decoders 810 are integrated into a combined codec architecture 820. Each processing module of the decoders 810 is factorized into a generic part and a specific part. The program memory for the generic decoding part can be shared between several voice coding standards, resulting in

smaller overall program size. Depending on the bitstream constraints, the number of codecs combined, and the similarity between the codecs combined, the decoder part 820 of the thin codec may achieve significant program size reductions. The bitstream constraints may include bit-exactness and minimum performance requirements.

5 [0047] Figure 9 is a simplified block diagram for an encoder of a thin CELP codec according to an embodiment of the present invention. This diagram is merely an example, which should not unduly limit the scope of the present invention. One of ordinary skill in the art would recognize many variations, alternatives, and modifications.

10 An encoder 900 of a thin CELP codec includes specific modules 990 and generic modules 992. The specific modules 990 include CELP encoding modules 920 and bitstream packing modules 940. The generic modules 992 include generic tables 960, generic math operations 970, and generic subfunctions 980. Input speech samples 910 are input to the codec-specific CELP encoding modules 920 and codec-specific CELP parameters 930 are produced. These parameters are then packed to a bitstream 950 in a desired coding standard format using the
15 codec-specific bitstream packing modules 940. The codec-specific CELP encoding modules 920 contain encoding modules for each supported voice coding standard. However, the tables 960, math operations 970 and subfunctions 980 that are common or generic to two or more of the supported encoders are factored out of the individual encoding modules by a codec algorithm factorization module, and included only once in a shared library in the thin
20 codec 900. This sharing of common code reduces the combined program memory requirements. Algorithm factorization is performed only once during the implementation stage for each combination of codecs in the thin codec. Efficient factorizing of subfunctions may require splitting the processing modules into more than one stage. Some stages may share commonality with other codecs, while other stages may be distinct to a particular
25 codec.

[0048] Figure 10 is a simplified block diagram for a decoder of a thin CELP codec according to an embodiment of the present invention. This diagram is merely an example, which should not unduly limit the scope of the present invention. One of ordinary skill in the art would recognize many variations, alternatives, and modifications. A decoder 1000 of a
30 thin CELP codec includes specific modules 1080 and generic modules 1090. The specific modules 1080 include bitstream unpacking modules 1020 and CELP decoding modules 1040. The generic modules 1090 includes generic tables 1050, generic math operations 1060, and

generic subfunctions 1070. A codec-specific bitstream 1010 is unpacked by the bitstream unpacking modules 1020, which contain a bitstream unpacking routine for each supported voice coding standard, and codec-specific CELP parameters 1030 are output to the CELP decoding modules 1040. The tables 1050, math operations 1060 and subfunctions 1070 that
5 are common or generic to more than two of the supported decoders are factored out of the codec-specific CELP decoding modules and included in a shared library.

[0049] The algorithm factorization module can operate at a number of levels depending on the codec requirements. If a bit-exact implementation is required to the individual standard codecs, only functions, tables, and math operations that maintain bit-exactness between more
10 than two codecs are factored out into the generic modules. Figure 11A is a simplified diagram showing generic modules between codec 1, codec 2 and code 3 for bit-exact implementation according to an embodiment of the present invention. This diagram is merely an example, which should not unduly limit the scope of the present invention. One of
15 ordinary skill in the art would recognize many variations, alternatives, and modifications. An area 1110 represents generic bit-exact modules of codecs 1, 2, and 3. Areas 1120, 1130, and 1140 represent generic bit-exact modules of codecs 1 and 3, codecs 1 and 2, and codec 2 and 3 respectively.

[0050] If the bit-exact constraint is relaxed, then functions, tables and math operations that produce equivalent quality or provide equivalent functionality can be factored out into the
20 generic modules. Alternatively, new generic processing modules can be derived and called by one or more codecs. This has the benefit of providing bit-compliant codec implementation. Using this approach, the program size can be reduced even further by having an increased number of generic modules. Figure 11B is a simplified diagram showing generic modules between codec 1, codec 2 and code 3 for equivalent performance
25 implementation according to an embodiment of the present invention. This diagram is merely an example, which should not unduly limit the scope of the present invention. One of ordinary skill in the art would recognize many variations, alternatives, and modifications. An area 1160 represents generic bit-exact modules of codecs 1, 2, and 3. Areas 1170, 1180, and 1190 represent generic bit-exact modules of codecs 1 and 3, codecs 1 and 2, and codec 2 and
30 3 respectively. For example, the area 1160 is larger than the area 1110, so more generic modules can be used in equivalent performance than in bit-exact implementation.

[0051] It is beneficial to maintain a modular, generalized framework so that modules for additional coders can be easily integrated. The use of generic modules may provide output voice quality higher than the standard codec implementation without an increase in program complexity, for example, by applying more advanced perceptual weighting filters. The use of generic modules may also provide lower complexity than the standard codec, for example, by applying faster searching techniques. These benefits may be combined.

[0052] The greater the similarity between voice coding standards, the greater the program size savings that can be achieved by a thin codec according to an embodiment of the present invention. As an example for illustration of the bit-compliant specific embodiment of a thin CELP codec, the speech codecs integrated are GSM-EFR, AMR-NB and AMR-WB, although others can be used. GSM-EFR is algorithmically the same as the highest rate of AMR-NB, thus no additional program code is required for AMR-NB to gain GSM-EFR bit-compliant functionality. The GSM standards AMR-NB, which has eight modes ranging from 4.75 kbps to 12.2 kbps, and AMR-WB, which has eight modes ranging from 6.60 kbps to 23.85 kbps, share a high degree of similarity in the encoder/decoder flow and in the general algorithms of many procedures.

[0053] According to one embodiment of the present invention, an apparatus for encoding and decoding a voice signal includes an encoder configured to generate an output bitstream signal from an input voice signal. The output bitstream signal is associated with at least a first standard of a first plurality of CELP voice compression standards. Additionally, the apparatus includes a decoder configured to generate an output voice signal from an input bitstream signal. The input bitstream signal is associated with at least a first standard of a second plurality of CELP voice compression standards. The output bitstream signal is bit exact or equivalent in quality for the first standard of the first plurality of CELP voice compression standards.

[0054] The CELP encoder includes a plurality of codec-specific encoder modules. At least one of the plurality of codec-specific encoder modules including at least a first table, at least a first function or at least a first operation. The first table, the first function or the first operation is associated with only a second standard of the first plurality of CELP voice compression standards. Additionally, the CELP encoder includes a plurality of generic encoder modules. At least one of the plurality of generic encoder modules includes at least a second table, a second function or a second operation. The second table, the second function

or the second operation is associated with at least a third standard and a fourth standard of the first plurality of CELP voice compression standards. The third standard and the fourth standard of the first plurality of CELP voice compression standards are different.

5 [0055] The plurality of codec-specific encoder modules includes a pre-processing module configured to process the speech for encoding, a linear prediction analysis module configured to generate linear prediction parameters, an excitation generation module configured to generate an excitation signal by filtering the input speech signal by the short-term prediction filter, and a long-term prediction module configured to generate open-loop pitch lag parameters. Additionally, the plurality of codec-specific encoder modules includes an
10 adaptive codebook module configured to determine an adaptive codebook lag and an adaptive codebook gain, a fixed codebook module configured to determine fixed codebook vectors and a fixed codebook gain; and a bitstream packing module. The bitstream packing module includes at least one bitstream packing routine and is configured to generate the output bitstream signal based on at least codec-specific CELP parameters associated with at least the
15 first standard of the first plurality of CELP voice compression standards.

[0056] The plurality of generic encoder modules comprises a first common functions library including at least the second function, a first common math operations library including at least the second operation, and a first common tables library including at least the second table. The first common functions library, the first common math operations
20 library and the first common tables library are made by at least an algorithm factorization module. The algorithm factorization module is configured to remove a first plurality of generic functions, a first plurality of generic operations and a first plurality of generic tables from the plurality of codec-specific encoder modules and store the first plurality of generic functions, the first plurality of generic operations and the first plurality of generic tables in
25 the first common functions library, the first common math operations library and the first common tables library.

[0057] The first common functions library, the first common math operations library and the first common tables library are associated with at least the third standard and the fourth standard of the first plurality of CELP voice compression standards and configured to
30 substantially remove all duplications between a first program code associated with the third standard of the first plurality of CELP voice compression standards and a second program

code associated with the fourth standard of the first plurality of CELP voice compression standards.

[0058] For example, the first common functions library, the first common math operations library and the first common tables library include only functions, math operations and tables configured to maintain bit exactness for the third standard and the fourth standard of the first plurality of CELP voice compression standards. For another example, the first common functions library, the first common math operations library and the first common tables library include only functions, math operations and tables algorithmically identical to ones of the third standard and the fourth standard of the first plurality of CELP voice compression standards, and functions, math operations and tables algorithmically similar to ones of the third standard and the fourth standard of the first plurality of CELP voice compression standards.

[0059] The CELP decoder includes a plurality of codec-specific decoder modules. At least one of the plurality of codec-specific decoder modules includes at least a third table, at least a third function or at least a third operation. The third table, the third function or the third operation is associated with only a second standard of the second plurality of CELP voice compression standards. Additionally, the CELP decoder includes a plurality of generic decoder modules. At least one of the plurality of generic decoder modules includes at least a fourth table, a fourth function or a fourth operation. The fourth table, the fourth function or the fourth operation is associated with at least a third standard and a fourth standard of the second plurality of CELP voice compression standards. The third standard and the fourth standard of the second plurality of CELP voice compression standards are different.

[0060] The plurality of codec-specific decoder modules include a bitstream unpacking module. The bitstream unpacking module includes at least one bitstream unpacking routine and is configured to decode the input bitstream signal and generate codec-specific CELP parameters. Additionally, the plurality of codec-specific decoder modules include an excitation reconstruction module configured to reconstruct an excitation signal based on at least information associated with adaptive codebook lags, adaptive codebook gains, fixed codebook indices and fixed codebook gains. Moreover, the plurality of codec-specific decoder modules include a synthesis module configured to filter the excitation signal and generate a reconstructed speech. Also, the plurality of codec-specific decoder modules

include a post-processing module configured to improve a perceptual quality of the reconstructed speech.

[0061] The generic decoder modules comprise a second common functions library including at least the fourth function, a second common math operations library including at least the fourth operation, and a second common tables library including at least the fourth table. The second common functions library, the second common math operations library and the second common tables library are made by at least an algorithm factorization module. The algorithm factorization module is configured to remove a second plurality of generic functions, a second plurality of operations and a second plurality of tables from the plurality of codec-specific decoder modules and store the second plurality of generic functions, the second plurality of operations and the second plurality of tables in the second common functions library, the second common math operations library and the second common tables library.

[0062] The second common functions library, the second common math operations library and the second common tables library are associated with at least the third standard and the fourth standard of the second plurality of CELP voice compression standards and configured to substantially remove all duplications between a third program code associated with the third standard of the second plurality of CELP voice compression standards and a fourth program code associated with the fourth standard of the second plurality of CELP voice compression standards.

[0063] For example, the second common functions library, the second common math operations library and the second common tables library include only functions, math operations and tables configured to maintain bit exactness for the third standard and the fourth standard of the second plurality of CELP voice compression standards. For another example, the second common functions library, the second common math operations library and the second common tables library include only functions, math operations and tables algorithmically identical to ones of the third standard and the fourth standard of the second plurality of CELP voice compression standards, and functions, math operations and tables algorithmically similar to ones of the third standard and the fourth standard of the second plurality of CELP voice compression standards.

[0064] As discussed above and further emphasized here, one of ordinary skill in the art would recognize many variations, alternatives, and modifications. For example, the first

plurality of CELP voice compression standards may be different from or the same as the second plurality of CELP voice compression standards. The first standard of the first plurality of CELP voice compression standards may be different from or the same as the first standard of the second plurality of CELP voice compression standards. The first standard of the first plurality of CELP voice compression standards may be different from or the same as the second standard of the first plurality of CELP voice compression standards. The first standard of the first plurality of CELP voice compression standards may be different from or the same as the third standard or the fourth standard of the first plurality of CELP voice compression standards. The first standard of the second plurality of CELP voice compression standards may be different from or the same as the second standard of the second plurality of CELP voice compression standards. The apparatus of claim 1 wherein the first standard of the second plurality of CELP voice compression standards is the same as the third standard or the fourth standard of the second plurality of CELP voice compression standards.

[0065] According to another embodiment of the present invention, a method for encoding and decoding a voice signal includes receiving an input voice signal, processing the input voice signal, and generating an output bitstream signal based on at least information associated with the input voice signal. The output bitstream signal is associated with at least a first standard of a first plurality of CELP voice compression standards. Additionally, the method includes receiving an input bitstream signal, processing the input bitstream signal, and generating an output voice signal based on at least information associated with the input bitstream signal. The output voice signal is associated with at least a first standard of a second plurality of CELP voice compression standards. The output bitstream signal is bit exact or equivalent in quality for the first standard of the first plurality of CELP voice compression standards. The output voice signal is bit exact or equivalent in quality for the first standard of the second plurality of CELP voice compression standards. For example, the first plurality of CELP voice compression standards include GSM-EFR, GSM-AMR Narrowband, and GSM-AMR Wideband. As another example, the first plurality of CELP voice compression standards includes EVRC and SMV.

[0066] The processing the input voice signal uses at least a first common functions library, at least a first common math operations library, and at least a first common tables library. The first common functions library includes a first function; the first common math operations library includes a first operation, and the first common tables library includes a first table. The first function, the first operation and the first table are associated with at least

a second standard and a third standard of the first plurality of CELP voice compression standards. The second standard and the third standard of the first plurality of CELP voice compression standards are different. The first common functions library, the first common math operations library and the first common tables library are made by at least an algorithm factorization module. The algorithm factorization module is configured to store a first plurality of generic functions, a first plurality of operations and a first plurality of tables in the first common functions library, the first common math operations library and the first common tables library.

[0067] The generating an output bitstream signal includes generating a first plurality of codec-specific CELP parameters based on at least information associated with the input voice signal, and packing the first plurality of codec-specific CELP parameters to the output bitstream signal. The first plurality of codec-specific CELP parameters include a linear prediction parameter, an adaptive codebook lag, an adaptive codebook gain, a fixed codebook index, and a fixed codebook gain. For example, the linear prediction parameter includes a line spectral frequency. The generating a first plurality of code-specific CELP parameters includes performing a linear prediction analysis, generating linear prediction parameters, and filtering the input speech signal by a short-term prediction filter. Additionally, the generating a first plurality of code-specific CELP parameters includes generating an excitation signal, determining an adaptive codebook pitch lag parameter, and determining an adaptive codebook gain parameter. Moreover, the generating a first plurality of code-specific CELP parameters includes determining an index of a fixed codebook vector associated with a fixed codebook target signal, and determining a gain of the fixed codebook vector.

[0068] The processing the input bitstream signal uses at least a second common functions library, at least a second common math operations library, and a second common tables library. The second common functions library includes a second function, the second common math operations library includes a second operation, and the second common tables library includes a second table. The second function, the second operation and the second table are associated with at least a second standard and a third standard of the second plurality of CELP voice compression standards. The second standard and the third standard of the second plurality of CELP voice compression standards are different.

[0069] The generating an output voice signal includes unpacking the input bitstream signal and decoding a second plurality of codec-specific CELP parameters to produce an output

voice signal. The decoding a second plurality of codec-specific CELP parameters includes reconstructing an excitation signal, synthesizing the excitation signal, and generating an intermediate speech signal. Additionally, the decoding a second plurality of codec-specific CELP parameters includes processing the intermediate speech signal to improve a perceptual
5 quality.

[0070] As discussed above and further emphasized here, one of ordinary skill in the art would recognize many variations, alternatives, and modifications. For example, the first plurality of CELP voice compression standards may be different from or the same as the second plurality of CELP voice compression standards. The first standard of the first
10 plurality of CELP voice compression standards is different from or the same as the first standard of the second plurality of CELP voice compression standards. The first standard of the first plurality of CELP voice compression standards may be different from or the same as the second standard or the third standard of the first plurality of CELP voice compression standards. The first standard of the second plurality of CELP voice compression standards
15 may be different from or the same as the second standard or the third standard of the second plurality of CELP voice compression standards.

[0071] Figure 12 is a simplified block diagram of an encoder for GSM-EFR and AMR-NB. GSM-EFR is algorithmically substantially the same as the highest rate of AMR-NB. Input speech samples 1210 is first preprocessed by a pre-processing module 1212, and 10th-order
20 linear prediction coefficients are determined once per frame or twice per frame for 12.2kbps mode by an LP windowing and autocorrelation module 1214 and a Levinson-Durbin module 1216. The Levinson-Durbin module 1216 uses the Levinson-Durbin algorithm. These 10th-order linear prediction coefficients are converted to line spectral frequencies (LSFs) by an LPC to LSF conversion module 1218. The converted frequencies are quantized by an LSF
25 quantization module 1220. The unquantized LSFs are interpolated by an LSF interpolation module 1222, and the quantized LSFs are interpolated by an LSF interpolation module 1224. These interpolated outputs are used in the computation of the weighted speech, impulse response and adaptive codebook target by modules 1226, 1228 and 1230 respectively. The open-loop pitch is determined from the weighted speech by a module 1232 and then refined
30 during the adaptive codebook search by a module 1234. The impulse response is computed and used in both the adaptive and fixed codebook searches. Once the adaptive lag is found, the adaptive codebook gain is determined, followed by the fixed codebook target, fixed codebook indices and fixed codebook gain. An ACELP fixed codebook structure is applied

for all modes. The codebook vectors are chosen by minimizing the error between the original signal and the synthesized speech using a perceptually weighted distortion measure.

[0072] Figure 13 is a simplified block diagram of an encoder for GSM AMR-WB. The encoder structure has a high degree of similarity to the AMR-NB structure. Input speech samples 1310 is first preprocessed in a pre-processing module 1312. The 16th-order linear prediction coefficients (LPCs) are determined once per frame using the Levinson-Durbin algorithm by an LP windowing and autocorrelation module 1314 and a Levinson-Durbin module 1316. The LPCs are converted to immittance spectral frequencies (ISFs) by an LPC to ISF conversion module 1318. The converted frequencies are quantized by an ISF quantization module 1320. The unquantized ISFs are interpolated by an ISF interpolation module 1322, and the quantized ISFs are interpolated by an ISF interpolation module 1324. These interpolated outputs are used in the computation of the weighting filter, impulse response and adaptive codebook target by modules 1326, 1328 and 1330. The open-loop pitch is determined from the weighted speech by a module 1332 and then refined during the adaptive codebook search by a module 1334. The impulse response is computed and used in both the adaptive and fixed codebook searches. One of two interpolation filters is selected for the fractional adaptive codebook search. Once the adaptive lag is found, the adaptive codebook gain is determined, followed by the fixed codebook target, fixed codebook indices and fixed codebook gain. An ACELP fixed codebook structure is applied for all modes. The codebook vectors are chosen by minimizing the error between the original signal and the synthesized speech using a perceptually weighted distortion measure. For a high rate, the gain of the high frequency range is determined and a gain index is transmitted.

[0073] A comparison of certain features and processing functions of AMR-NB and AMR-WB according to an embodiment of the present invention is shown in Table 1. This table is merely an example, which should not unduly limit the scope of the present invention. One of ordinary skill in the art would recognize many variations, alternatives, and modifications.

	AMR-NB	AMR-WB
Frame size	20ms	20ms
Subframes per frame	4	4
Sampling rate	8 kHz	16 kHz
Pre-processing	Highpass filtering (80Hz)	Upsample by 4, LPF 6.4kHz, Downsample by 5 Highpass filtering (50Hz) Pre-emphasis $H(z)=1-0.68z^{-1}$

LP analysis	10 th order LP analysis LPC to LSP conversion	16 th order LP analysis LPC to ISP conversion
LP param. Quant.	Quantize LSFs Split matrix quantization (SMQ) or Split Vector Quantization (SVQ)	Quantize ISPs Split Multi-stage vector quantization, 2 stages
Weighting filter	$W(z) = A(z/\gamma_1)/A(z/\gamma_2)$	$W(z) = A(z/\gamma_1) / (1-0.68z^{-1})$
Open-loop pitch	Pitch lag range 18 - 143 Use 3 ranges or weighting function	Pitch lag range 17 - 115 Use a weighting function
Closed-loop pitch	Adaptive codebook Range 17,19 - 143 1/6, 1/3 sample resolution	Adaptive codebook Range 34 - 231 1/2, 1/4 sample resolution
Fixed codebook structure and search	ACELP, 40 samples / subframe Different tracks and no. of pulses for each mode. adaptive prefilter $F(z) = 1/(1-g_p z^{-T})$	ACELP, 64 samples / subframe Different no. of pulses for each mode adaptive prefilter $F(z) = 1/(1-0.85 z^{-T}) (1-b_1 z^{-1})$
Gain quantization	Joint VQ with 4 th order MA prediction or Separate quantization of gc, gp	Joint VQ with 4 th order MA prediction
High band frequency	n/a	Transmit high-band gain for highest rate Generate 6.4-7kHz with scaled white noise, convert to speech domain.
Post-processing	Adaptive tilt compensation filter Formant postfilter Highpass filtering	Highpass filtering De-emphasis filter Upsample by 5, Downsample by 4

Table 1

[0074] As shown in Table 1, both AMR-NB and AMR-WB operate with a 20 ms frame size divided into 4 subframes of 5 ms. A difference between the wideband and narrowband coder is the sampling rate, which is 8 kHz for AMR-NB and 16 kHz downsampled to 12.8 kHz for analysis for AMR-WB. AMR wideband contains additional pre-processing functions for decimation and pre-emphasis. The linear prediction (LP) techniques used in both AMR-NB and AMR-WB are substantially identical, but AMR-WB performs linear prediction (LP) analysis to 16th order over an extended bandwidth of 6.4 kHz and converts the LP coefficients to/from Immittance Spectral Pairs (ISP). Quantization of the ISPs is performed using split-multi-stage vector quantization (SMSVQ), as opposed to split matrix quantization and split vector quantization for quantization of the LSFs in AMR-NB. The pitch search routines and computation of the target signal are similar, although the sample resolution for pitches differs. Both codecs follow an ACELP fixed codebook structure using a depth-first tree search to reduce computations. The adaptive and fixed codebook gains are quantized in both codecs using joint vector quantization (VQ) with 4th order moving average (MA) prediction. AMR-NB also uses scalar gain quantization for some modes. AMR-WB contains

additional functions to deal with the higher frequency band up to 7 kHz. The post-processing for both coders includes high-pass filtering, with AMR-NB including specific functions for adaptive tilt-compensation and formant postfiltering, and AMR-WB including specific functions for de-emphasis and up-sampling.

- 5 **[0075]** Figure 14 is a simplified block diagram for an encoder of a thin codec for GSM-EFR, AMR-NB and AMR-WB according to an embodiment of the present invention. This diagram is merely an example, which should not unduly limit the scope of the present invention. One of ordinary skill in the art would recognize many variations, alternatives, and modifications. Modules 1410 and 1412 for LP analysis, modules 1414 and 1416 for
10 interpolation, a module 1418 for open-loop pitch search, modules 1420 and 1422 for adaptive and fixed target computation respectively, and a module 1424 for impulse response computation have a high degree of similarity and can be generic without substantial loss of quality. The modules 1410 and 1412 for LP analysis may include a module 1410 for autocorrelation and a module 1412 for Levinson-Durbin. The modules of computing
15 weighted speech, closed-loop pitch search, ACELP codebook search, search and construct excitation also contain similarity in the processing, although conditions and parameters may vary. For example, the search methods for the ACELP fixed codebook can be shared, but the algebraic structures differ. The quantization modules are mostly codec-specific and the high-band processing functions are usually used only by AMR-WB.
- 20 **[0076]** Figure 15 is a simplified block diagram for a decoder of a thin codec for GSM-EFR AMR-NB and AMR-WB according to an embodiment of the present invention. This diagram is merely an example, which should not unduly limit the scope of the present invention. One of ordinary skill in the art would recognize many variations, alternatives, and modifications. Modules 1524, 1510, 1512, and 1514 for interpolation, excitation
25 reconstruction, synthesis and post-processing respectively have a high degree of similarity and can be generic without substantial loss of quality. Bitstream decoding modules 1516 and 1518 are codec-specific. The adaptive codebook filter 1520 and high-band processing functions 1522 are usually used only for AMR-WB. At least some generic modules are shared between the codecs. Additionally, common tables, subfunctions and operations of
30 codec-specific modules are also factorized out into a shared library to further reduce the program size.

[0077] As another example for illustration of the bit-compliant specific embodiment, a thin CELP codec is applied to integrate the Code Division Multiple Access (CDMA) standards SMV and EVRC, although others can be used. SMV has 4 bit rates including Rate 1, Rate $\frac{1}{2}$, Rate $\frac{1}{4}$ and Rate $\frac{1}{8}$ and EVRC has 3 bit rates including Rate 1, Rate $\frac{1}{2}$ and Rate $\frac{1}{8}$.

5 [0078] Figure 16 is a simplified block diagram for an encoder for EVRC. A signal 1610 is passed to a pre-processing module 1612 which performs highpass filtering to suppress very low frequencies and noise reduction to lessen background noise. Linear prediction analysis is performed by a module 1614 once per frame using the Levinson-Durbin recursion producing autocorrelation coefficients and linear prediction coefficients (LPCs). The LPCs are
10 converted to LSPs by a module 1616 and interpolated by a module 1618. The excitation is generated by a module 1620 that performs inverse filtering of the pre-processed speech by the inverse linear prediction filter. The open-loop pitch lag and pitch gain are then estimated. Using the autocorrelation coefficients, the pitch gain, and an external rate command, the bit rate for the current frame is determined by a module 1622. The rate determination module
15 1622 applies voice activity detection (VAD) and logic operations to determine the rate. Depending on the bit rate, a different processing path is selected. For Rate $\frac{1}{8}$, the parameters transmitted are the LSPs, quantized to 8 bits, and the frame energy. For Rate $\frac{1}{2}$ and Rate 1, the LSPs, pitch lag, adaptive codebook gain, fixed codebook indices and fixed codebook gains are computed. Rate 1 has the additional parameters of spectral transition
20 indicator and delay difference. The LSFs are quantized first and RCELP processing is performed, whereby the signal is modified by time-warping so that the signal has a smooth pitch contour. The adaptive and fixed codebook vectors are selected to match the modified speech signal.

[0079] Figure 17 is a simplified block diagram of the encoder for SMV. A signal 1710 is
25 passed to a pre-processing module 1712 which performs silence enhancement, highpass filtering, noise reduction and adaptive tilt filtering. Linear prediction analysis is performed by a module 1714 three times per frame, centered at different locations, using the Levinson-Durbin recursion producing autocorrelation coefficients and linear prediction coefficients (LPCs). The LPCs are converted to LSPs by a module 1716. The pre-processed speech is
30 perceptually weighted, and the open-loop pitch lag and frame class/type are estimated. The lag is used to modify the pre-processed speech by time-warping and the frame class may be updated. Using numerous analysis parameters, including the frame class, the bit rate for the current frame is determined. Depending on the bit rate and frame type, a different processing

path is selected. For Rate 1/8, the parameters transmitted are the LSPs, quantized to 11 bits, and the subframe gains. For Rate 1/4, noise excited linear prediction (NELP) processing is performed. For Rate 1/2 and Rate 1, two processing paths are available for each rate, Type 1 and Type 0. In each case, the LSPs, LSP predictor switch, adaptive codebook lags, adaptive codebook gain, fixed codebook indices and fixed codebook gains are computed. Rate 1, Type 0 has the additional parameter of LSP interpolation path. The LSFs are quantized first and either CELP (Type 0) or RCELP (Type 1) processing is performed, whereby the signal is modified by time-warping so that the signal has a smooth pitch contour.

[0080] A comparison of certain features and processing functions of SMV and EVRC according to an embodiment of the present invention is shown in Table 2. This table is merely an example, which should not unduly limit the scope of the present invention. One of ordinary skill in the art would recognize many variations, alternatives, and modifications.

	SMV	EVRC
Frame size	20ms	20ms
Subframes per frame	4, 3, or 2 depending on Rate and Frame type	3 (53, 53, 54 samples)
Sampling rate	8 kHz	8 kHz
Pre-processing	Silence enhancement High-pass filtering (80Hz, 2 nd order) Noise pre-processing (2 options) Adaptive Tilt filter	Highpass filtering (120Hz, 6th order) Noise pre-processing (same as SMV option A)
LP analysis	10 th order LP analysis LPC to LSP conversion	10 th order LP analysis LPC to LSP conversion
Rate Selection / VAD	Rate based on input characteristics 2 VAD options	Rate based on input characteristics (Rate determination identical to one of SMV VAD options)
LSP Quant.	Switched MA prediction, 2 predictors Weighted Multi-stage VQ (MSVQ)	Weighted Split Vector Quantization (SVQ)
Pitch search	Integer and fractional delay search on weighted speech	Integer pitch search on residual No closed- loop search
Target signal	RCELP signal modification Warp/Shift weighted speech to match pitch contour	RCELP signal modification Shift residual to match pitch contour
Fixed codebook	ACELP and Gaussian codebooks Iterative depth-first tree search	ACELP codebooks Iterative depth-first search or exhaustive search
Gain quantization	Joint quantization of adaptive and fixed gains	Separate quantization of adaptive and fixed gains
Low rates	NELP processing for Rate 1/4 Gaussian excitation for Rate 1/8	Gaussian excitation for Rate 1/8
Post processing	Tilt compensation Formant post-filter Long-term postfilter Highpass filtering	Formant postfilter Highpass filtering

Table 2

[0081] As shown in Table 2, SMV and EVRC share a high degree of similarity. At the basic operations level, SMV math functions are based on EVRC libraries. At the algorithm level, both codecs have a frame size of 20ms and determine the bit rate for each frame based on the input signal characteristics. In each case, a different coding scheme is used depending on the bit rate. SMV has an additional rate, Rate ¼, which uses NELP encoding. The noise suppression and rate selection routines of EVRC are identical to SMV modules. SMV contains additional preprocessing functions of silence enhancement and adaptive tilt filtering. The 10th order LP analysis is common to both codecs, as is the RCELP processing for the higher rates which modifies the target signal to match an interpolated delay contour. Both codecs use an ACELP fixed codebook structure and iterative depth-first tree search. SMV also uses Gaussian fixed codebooks. At Rate 1/8, both codecs produce a pseudo-random noise excitation to represent the signal. SMV incorporates the full range of post-processing operations including tilt compensation, formant postfilter, long term postfilter, gain normalization, and highpass filtering, whereas EVRC uses a subset of these operations.

[0082] Figure 18 is a simplified block diagram of an embodiment of an encoder of a thin codec for SMV and EVRC according to an embodiment of the present invention. This diagram is merely an example, which should not unduly limit the scope of the present invention. One of ordinary skill in the art would recognize many variations, alternatives, and modifications. A module 1810 for LP analysis, a module 1812 for LPC to LSP conversion, a module 1814 for perceptual weighting, a module 1816 for open-loop pitch search, a module 1818 for RCELP modification, and module 1820 for generating random excitation have a high degree of similarity and can be generic. The module 1810 may perform autocorrelation and Levinson-Durbin processing. Additionally, modules for interpolation, adaptive and fixed target computation, and impulse response computation also have a high degree of similarity and can be generic. The Rate 1/8 processing is similar to both SMV and EVRC codecs, while the Rate 1 and Rate ½ processing of EVRC is similar to Type 1 SMV processing. SMV requires additional classification processing to accurately classify the input, and additional processing paths to accommodate both Type 1 and Type 0 processing. Many of the fixed codebook search functions are generic as both codecs include ACELP codebooks. Since SMV is considerably more algorithmically complex than EVRC, a possible approach for one or more of the thin codec encoding modules, for example the rate determination module, is to embed EVRC functionality within the SMV processing modules. These modules may be split into stages, with some stages generic to each codec. Other modules

containing some generic stages include module 1822 for pre-processing, and module 1824 for rate determination.

[0083] Figure 19 is a simplified block diagram of an embodiment of a decoder of a thin codec for SMV and EVRC according to an embodiment of the present invention. This diagram is merely an example, which should not unduly limit the scope of the present invention. One of ordinary skill in the art would recognize many variations, alternatives, and modifications. Similar to the encoder as shown in Figure 18, there are different processing paths, depending on the bit rate. The bitstream decoding modules are codec-specific and the post-processing operations for EVRC can be embedded within the SMV post-processing module. Module 1910 for Rate 1/8 decoding has a high degree of similarity and can be generic. In addition to shared decoding modules, common tables, subfunctions and operations of codec-specific modules are also factorized out into a shared library to further reduce the program size.

[0084] As discussed above and further emphasized here, Figures 18 and 19 are merely examples. The apparatus and method for a thin CELP voice codec is applicable to numerous combinations of various voice codecs. For example, these voice codecs include G.723.1, GSM-AMR, EVRC, G.728, G.729, G.729A, QCELP, MPEG-4 CELP, SMV, AMR-WB, and VMR. Usually, the more similar the codec algorithms, the greater the potential achievable program size savings.

[0085] Numerous benefits are achieved using the present invention over conventional techniques. Certain embodiments of the present invention can be used to reduce the program size of the encoder and decoder modules to be significantly less than the combined program size of the individual voice compression modules. Some embodiments of the present invention can be used to produce improved voice quality output than the standard codec implementation. Certain embodiments of the present invention can be used to produce lower computational complexity than the standard codec implementation. Some embodiments of the present invention provide efficient embedding of a number of standard codecs and facilitate interoperability of handsets with diverse networks.

[0086] Although specific embodiments of the present invention have been described, it will be understood by those of skill in the art that there are other embodiments that are equivalent to the described embodiments. Accordingly, it is to be understood that the invention is not to

be limited by the specific illustrated embodiments, but only by the scope of the appended claims.